เมธี เผ่าบุญมี

การพัฒนาระบบ ซอฟต์แวร์

กล่าวถึงการพัฒนาของระบบซอฟต์แวร์และปัญหาต่าง ๆ ที่เกิดขึ้นในปัจจุบัน

งานที่บริษัทผมทำอยู่คือการเป็นที่ปรึกษาและพัฒนาระบบคอมพิวเตอร์ให้กับองค์กรต่าง ๆ ก็อยากจะนำปัญหาที่พบในการทำงานมาเล่าสู่กันฟังแบบสบาย ๆ และในหลาย ๆ มุมมอง ปัญหาส่วนใหญ่จะเป็นเรื่องของเทคโนโลยีเฟ้อ คือเทคโนโลยีทางด้านคอมพิวเตอร์ที่เกิดใหม่ยังไม่สุกดี หมายถึง ยังเป็นกระดาษของนักวิชาการก็มี ที่สุก ๆ ดิบ ๆ นำมาใช้งานแล้วบางส่วนก็มี ที่สุกงอมนำมาใช้แล้วเต็มที่ก็มาก รวม ๆ แล้วที่เราได้ยินจากสื่อต่าง ๆ ทุกวันก็มากจนไม่รู้ว่าจะเอาไปใช้ยังไงดี แม้มีเงินซื้อก็ยังซื้อไม่ถูก บริษัทที่ปรึกษาอย่างผมก็ต้องทำหน้าที่แจกแจงว่าสิ่งที่เราจะทำให้ลูกค้านะมันดีอย่างไร

รายชื่อบางส่วนของเทคโนโลยีของระบบคอมพิวเตอร์ที่กำลังเป็นที่กล่าวถึงอยู่ในปัจจุบันคือ

System
- **Client-Server system**
- **Distributed system**
- **Open system**
- **Peer-to-peer system**
- **Workgroup system**

Database
- **Client-Server database**
- **Distributed database**
- **Multi database**
- **Open database**

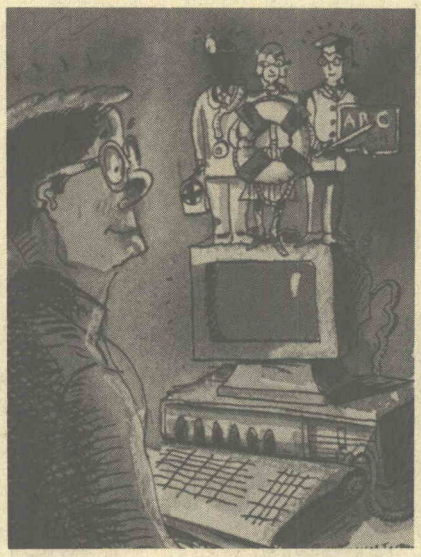
Networking

LAN
Ethernet
Token-Ring
ทางด่วนข้อมูล
TCP/IP
Novell Netware
Multimedia
Object Oriented
Database
Design
Programming
Windows
ฯลฯ

วันนี้ผมคงจะคุยกันถึงเรื่องทั่ว ๆ ไปในการพัฒนาซอฟต์แวร์ ซึ่งเป็นการปรับความเข้าใจให้ตรงกันก่อนที่จะคุยถึงเรื่องที่เฉพาะเจาะจงต่อไป

ตัดสินใจผิดก็ต้องคิดหนัก

ผมพบว่าฝ่ายคอมพิวเตอร์ขององค์กรมักติดปัญหาที่ว่าไม่ค่อยมีเวลามาศึกษาเทคโนโลยีใหม่ ๆ กัน เหตุผลก็คือยุ่งกับงานประจำวันมากเกินไป เพราะหลังจากที่ติดตั้งระบบเสร็จแล้วก็มัวแต่แก้ปัญหาทั้งของระบบและของผู้ใช้ในแต่ละวันก็หมดเวลาแล้ว บางที่ต้องคอยแก้ปัญหาตั้งแต่ระดับเมนเฟรมจนถึงไมโครฯ (เนื่องจากเอาไมโครฯไปใช้ทำเป็นเทอร์มินัลของเมนเฟรมด้วย ก็เลยมีทั้งสองปัญหาปนกัน) อย่างไรก็ตาม ก็ขอให้ช่วย ๆ กันตามหน่อยนะครับ ถ้าตามไม่ทันจะเกิดผลเสียมาก เพราะว่าเทคโนโลยีใหม่ ๆ เกิดขึ้นและเปลี่ยนไปอย่างรวดเร็วมาก ในขณะเดียวกันองค์กรปัจจุบันก็มีขนาดใหญ่และซับซ้อนมากขึ้น ถ้ามัวแต่ยุ่งกับงานประจำวัน สักวันหนึ่งคุณกับผม (ที่ปรึกษา) อาจจะคุยกันไม่รู้เรื่องไปเลยก็ได้ ผมก็จะไม่มีงานทำนะซิครับ อย่าบอกนะครับว่าคุณสามารถซื้อแบบเหมารวบยอด (Turnkey) ได้ ระวังว่าคุณอาจจะได้เทคโนโลยีแบบจูราสสิกแทน ถ้าเจอชนิดที่ยังไม่ทันติดตั้งเสร็จก็ล้าสมัยไปแล้ว แบบนี้ก็ต้องคิดหนักแน่นอน เพราะจะขยับขยายอะไรก็ลำบาก



ตัวอย่างที่ผมพบมาเป็นระบบที่ติดตั้งอยู่ก่อนมาสัก 2-3 ปีเท่านั้น บริษัทของผมต้องการขอเชื่อมกับฐานข้อมูลของระบบเพื่อไปออกรายงานสรุปให้ผู้บริหารขององค์กร อยากให้การเชื่อมต่อเป็นแบบไร้ตะเข็บเลย ถามผู้ขายว่ามีวิธีไหนบ้างจริง ๆ แล้วผมพอทราบมาว่าเป็นระบบปิดสนิท ก็ทำใจเอาไว้ว่ากรณีที่แย่ที่สุดก็คือต้องขอข้อมูลมาเป็นแบบเท็กซ์ไฟล์วันต่อวันมา Update ในระบบใหม่ ซึ่งเป็นความลำบากอย่างหนึ่งที่ต้องทำข้อมูลให้ตรงกันเสมอ ความหวังที่จะใช้สิ่งใหม่ ๆ เข้าช่วย เช่น หา Gateway หรือใช้ ODBC มีน้อยมาก หรือถ้ามีก็คงแพงมาก คำตอบที่ได้ผิดความคาดหมายไปมากพอดู คือฐานข้อมูลตัวนี้ไม่มีการออกรุ่นใหม่ ๆ ไม่มีโปรแกรมเสริมอื่น ๆ มานานแล้ว เรียกว่าตายสนิทมานานแล้ว ที่ร้ายกว่านั้นคือ แม้เพียงแค่ต้องการข้อมูลที่เป็นเท็กซ์ไฟล์ก็ยังต้องไปจ้างผู้ขายเดิมให้ทำให้ ก็ปวดหัวกันทั้งสองฝ่าย

ปัญหานี้มีข้อสังเกตอยู่อย่างหนึ่งคือ มีผู้ทำนายไว้ว่าต่อไปฝ่ายคอมพิวเตอร์ในองค์กรจะตามเทคโนโลยีไม่ทัน และจะเกิดปัญหามาก เท่าที่เป็นอยู่ผมคิดว่ามีความเป็นไปได้มากทีเดียว ผลที่เกิดขึ้นจะเป็นอย่างไรนั้น ผมจะมาคุยให้ฟังในโอกาสต่อไป

เทคโนโลยีช็อค

ก็เป็นคำฮิตในยุคโลกาภิวัฒน์ละครับ สาเหตุเนื่องมาจากว่าเราใช้เทคโนโลยีสมัยเมื่อ 10 ปีก่อนอยู่ดี ๆ เผลอนิดเดียวก็มีเทคโนโลยีของปัจจุบันและของอนาคตมาให้เราเลือกใช้เต็มไปหมด เทคโนโลยีต่าง ๆ จากแหล่งกำเนิดที่สุกงอมแล้ว (เรียกว่าขายได้แล้วนั่นเอง) ก็แห่กันมารุมตอมประเทศที่มีอัตราการเติบโตทางเศรษฐกิจสูงอย่างรวดเร็ว ลูกค้าบางท่านที่ยังทำระบบรวมศูนย์อยู่พอผมเล่าให้ฟังว่าตอนนี้มีอะไรบ้างและต่อไปน่าจะเป็นอย่างไร เขาก็ทำท่าหนักใจว่ามันเปลี่ยนเร็วเหลือเกิน ผมก็บอกเขาว่ายังไม่ต้องหนักใจหรอก เพราะตอนนี้สินค้าสำหรับนำมาประกอบเป็นระบบที่ขายกันอยู่ก็ยังเปลี่ยนไปไม่มาก ยังมีส่วนที่เป็นของเก่าเอามาจัดใหม่อยู่เยอะ ส่วนจะเอามาจัดอย่างไรนั้นเดี๋ยวค่อยว่ากัน

แต่ที่แน่ ๆ ก็คือ ทางผู้ผลิตก็ต้องพยายามสร้างสิ่งที่ดีกว่ามาให้กับผู้ใช้อยู่แล้ว ดังนั้นขอให้มองในแง่ดีไว้ก่อนว่า เจ้าสิ่งใหม่ ๆ เหล่านี้จะมาช่วยลดงานของเราลงในอนาคต แต่เราก็ต้องออกแรงคือ ต้องกล้าที่จะนำมาลองใช้ กล้าที่จะเปลี่ยนแปลง โลกปัจจุบันนี้เปลี่ยนแปลงเร็ว ไม่

ค่อยมีอะไรอยู่คงที่ ของใหม่ที่ดีกว่าก็มาแทนของเก่า ถ้าเรามัวแต่ยึดว่าแบบนี้แหละดีแล้ว โอกาสที่จะล้าหลังก็มีสูง อีกอย่างที่ต้องออกแรงก็คือ ต้องใช้เวลาศึกษาสักเล็กน้อย ถ้าอย่างไหนใช้เวลาศึกษาแล้วเข้าใจยากแสดงว่าสิ่งนั้นไม่ดีจริง โยนทิ้งได้ ผมเองขอยืนยัน นั่งยัน และนอนยันเลยครับว่า ทางผู้ผลิตต้องทำให้สินค้าของเขาเข้าใจง่ายอย่างเต็มที่อยู่แล้ว ทั้งในด้านประโยชน์ที่จะได้รับและการใช้งาน มิฉะนั้นเขาก็ขายไม่ออกซิครับ

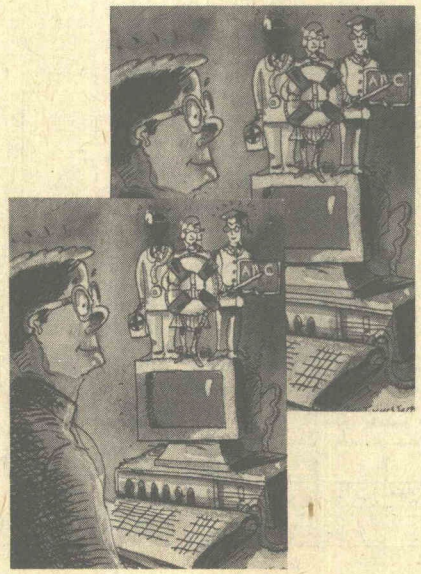
ซอฟต์แวร์กับจิ๊กซอ

สิ่งที่ผมสังเกตในเรื่องของการพัฒนาระบบซอฟต์แวร์ หรือการจะสร้างโปรแกรมขึ้นมาใช้งานนั้น ความก้าวหน้าที่จะช่วยให้การสร้างโปรแกรมขึ้นใหม่แบบง่าย ๆ ยังมีไม่มาก นักวิชาการเองก็ยังไม่มี "กฎ" ที่แน่นอน มีเพียงแค่ "วิธีการ" พูดง่าย ๆ ก็คือ ยังไม่เป็นวิทยาศาสตร์เต็มขั้น ยังมีส่วนที่ต้องใช้ศิลปเฉพาะตัวกันอยู่ ถึงแม้ตอนนี้จะมีโปรแกรมที่ใช้สร้างโปรแกรมพวก CASE มาใช้กันแล้ว แต่ก็ยังมีราคาแพงและยังไม่ทราบว่าของใครจะดีที่สุด

ผมยังมองว่าการเปลี่ยนแปลงในปัจจุบันถึงแม้ว่าจะเปลี่ยนไปอย่างรวดเร็ว แต่ก็ยังไม่เปลี่ยนขนาดหน้ามือเป็นหลังมือ บุคลากรที่ทำงานอยู่ในสาขาคอมพิวเตอร์ยังสามารถที่จะทำความเข้าใจได้ ทั้งนี้ก็เนื่องจากว่าการสร้างโปรแกรมนั้นเป็นเรื่องที่มีรายละเอียดมาก จะค้นหาวิธีการที่ดี ๆ มาใช้ในทันทีทันใดทำได้ไม่ง่ายนัก เมื่อคิดอะไรที่พอช่วยได้ก็นำออกมาใช้ก่อน รายละเอียดที่ย่อยที่สุดก็คือ โค้ดหนึ่งบรรทัด ในระบบใหญ่ ๆ ก็อาจจะมีจำนวนบรรทัดของโค้ดเป็นหมื่น ๆ แสน ๆ บรรทัด ถ้าคุณเคยเล่นต่อภาพหรือจิ๊กซอลองนึกถึงการเอาชิ้นส่วนเป็นหมื่นชิ้นมาต่อกันให้เป็นรูปว่ามันยากขนาดไหน ซึ่งในสมัยก่อนหลักเกณฑ์ว่าควรจะเอาโค้ดเหล่านั้นมาจัดอย่างไรก็มีเพียงให้มันทำงานตามที่เราต้องการเท่านั้น ไม่ต้องห่วงว่าในอนาคตจะต้องมีคนอื่นมาแก้ไขระบบหรือไม่ เรื่องการจัดเจ้าจิ๊กซอนี้ก่อความยากลำบากมาก ทั้งในด้านที่จะสร้างระบบขึ้นมาและในด้านที่จะรักษาไว้ให้คงอยู่นาน ๆ

ปัญหาจิ๊กซอของโค้ด

ถ้าจะมองความพยายามทำให้การพัฒนาระบบมีประสิทธิภาพมากขึ้นนั้น ผมมองว่าหลักใหญ่ ๆ ก็เพื่อมาจัดการกับเจ้าจิ๊กซอนั่นเอง เริ่มตั้งแต่การมีภาษาระดับสูงซึ่งทำให้จิ๊กซอแต่ละชิ้นที่ผู้พัฒนาระบบจริง ๆ มองเห็นมีขนาดใหญ่ขึ้น ผลก็คือ จำนวนชิ้นและความซับซ้อนจะน้อยลง การอนุญาตให้สร้างสัปรูทีนได้ก็เป็นการยอมให้ผู้พัฒนาระบบจัดกลุ่มของจิ๊กซอขึ้นมาเองได้ เปรียบเหมือนกับแทนที่จะต่อภาพทั้งภาพเลย ก็ยอมให้ต่อแค่เฉพาะภูเขาหรือต้นไม้ก่อน หลังจากนั้นค่อยเอามาต่อในภาพใหญ่อีกที ฯลฯ



เทคโนโลยีในปัจจุบันนี้เราก็ยังคงยุ่งอยู่กับเรื่องการจัดเจ้าจิ๊กซอของโค้ดอยู่ ตรงนี้แหละที่ผมบอกว่าไม่ต้องหนักใจกับเรื่องเทคโนโลยีให้มากนัก ถ้าจินตนาการว่าเรามีระบบเก่ารุ่นจูราสสิกอยู่ระบบหนึ่งเขียนด้วยโคบอลทั้งหมด มีโค้ดอยู่ประมาณหมื่นบรรทัด เราต้องการสร้างระบบนั้นใหม่ทั้งหมดโดยใช้เทคโนโลยีที่ก้าวหน้าที่สุดที่ขายกันอยู่ในปัจจุบัน เอาแค่ไหนดี ลองให้เป็นไคล์เอ็นต์เซอร์ฟเวอร์ ต้อง Distributed ผ่านเน็ตเวอร์กต้องเป็นระบบเปิด และให้เป็น GUI อีกด้วย แค่นี้คงพอนะครับ

ทีนี้พอสร้างเสร็จแล้วลองเอาทั้งสองระบบมาเปรียบเทียบกันดู เราจะพบว่าส่วนประกอบหลัก ๆ ยังคงอยู่ วิธีการเขียนโปรแกรมยังคล้าย ๆ แบบเดิม เพียงแต่จัดกลุ่มของจิ๊กซอใหม่ เช่น โค้ดส่วนการจัดการเกี่ยวกับข้อมูลของโคบอลก็ถูกรวมมาอยู่ที่ฐานข้อมูล โค้ดส่วนการสำรองข้อมูลฐานข้อมูลก็ทำให้ โค้ดส่วนที่เกี่ยวกับการติดต่อกับผู้ใช้ก็ยกให้เป็นเรื่องของวินโดวส์ ซึ่งการเขียนโปรแกรมก็เริ่มต่างไปจากเดิมบ้าง

สิ่งที่ทำให้เรารู้สึกยุ่งยากสับสนก็คือ ระบบจะมีส่วนประกอบมากขึ้น มีภาษาหลายภาษาแทนที่จะทำอะไรก็ใช้โคบอลหมด จะติดต่อกับฐานข้อมูลสักทีอาจจะต้องผ่านเครือข่ายที่ต้องมีสิ่งที่เรียกว่าโปรโตคอลวิ่งกันให้วุ่นวายในสาย ตรงนี้แหละครับที่บางคนจะรู้สึกว่างง แต่ถ้าเราเอาจำนวนโค้ดที่นักพัฒนาระบบต้องเขียนมาเทียบกันจะเห็นว่าระบบใหม่จะเขียนน้อยกว่า เนื่องจากโค้ดส่วนที่เหลือเรา "ซื้อ" มาแทนในรูปของโปรแกรมฐานข้อมูล ในรูปของวินโดวส์ ในระบบใหม่เราจะได้ความสามารถพิเศษมากกว่า เช่น มีการติดต่อกับผู้ใช้ที่สวยงาม มีระบบรักษาความถูกต้องของข้อมูลที่แน่นอนผ่าน

การทดสอบมาแล้ว แรงที่ต้องลงไปในการสร้างโดยใช้เทคโนโลยีใหม่จะพอ ๆ กัน หรือมากกว่าการสร้างโดยเทคโนโลยีเก่าก็เฉพาะในระบบแรก ๆ เท่านั้น ถ้าชำนาญแล้วงานของผู้พัฒนาระบบจะน้อยลงแถมยังได้ระบบที่ดีกว่าด้วย ที่สำคัญแนวโน้มของเทคโนโลยีจะเป็นแบบ "เปิด" มากขึ้น เมื่อเราศึกษาและนำมาใช้ก็จะทำให้เราสามารถ "ถอด" ส่วนต่าง ๆ ของระบบออกมา เอาของใหม่ที่ดีกว่าใส่เข้าไปแทนได้ง่าย ซึ่งจะช่วยในการรักษาอายุของระบบให้ยืนยาว

ไคล์เอ็นต์เซอร์ฟเวอร์ เรื่องเก่าเอามาจัดใหม่

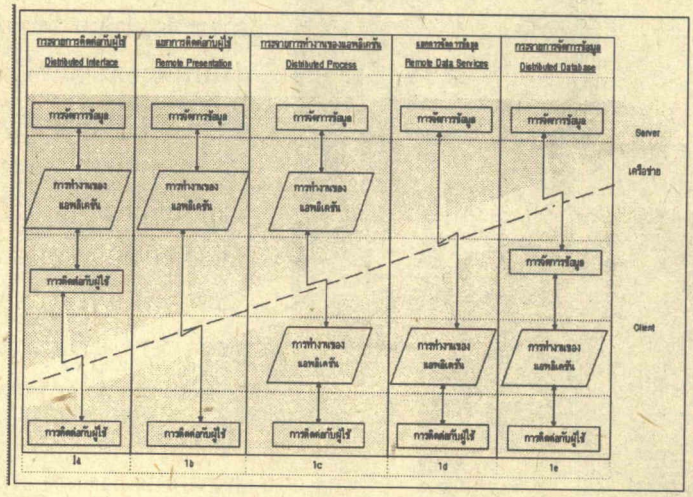
สมัยที่ผมยังทำงานอยู่กับเครื่อง PC และระบบรวมศูนย์ (จริง ๆ ตอนนั้นคงยังไม่มีคำว่ารวมศูนย์ แต่เรียกเป็นระบบมินิ เมนเฟรม) เมื่อได้ยินคำว่ากระจายศูนย์ (Distributed) ก็นึกว่าคงต้องใช้กับระบบใหญ่ ๆ เราคงไม่มีโอกาสจะได้ทำ แต่พอไปเรียนต่อถึงได้รู้ว่างานในระบบมินิหรือเมนเฟรมที่ทำกันอยู่ก็เรียกได้ว่าเป็นประมวลผลแบบกระจายศูนย์เหมือนกัน แต่มี "ความกระจายศูนย์" น้อยที่สุด ที่เป็นเช่นนี้เพราะความหมายง่าย ๆ ของการกระจายศูนย์คือ มีเครื่องหลายเครื่องแบ่งงานกันทำ ในกรณีของระบบเมนเฟรมก็มีเทอร์มินัลช่วยเมนเฟรมทำอยู่ แต่เมื่อเทียบกับปริมาณงานที่เมนเฟรมทำก็เทียบกันไม่ติด ในแง่การค้าแล้วไม่มีบริษัทใดเรียกว่าเป็นระบบกระจายศูนย์ รูปที่ 1 แสดงวิธีการแยกการประมวลผลแบบต่าง ๆ ซึ่งมีความกระจายศูนย์แตกต่างกันไป

รูปนี้แสดงให้เห็นเรื่องการเอาของเก่ามาจัดใหม่ได้เป็นอย่างดี ระบบซอฟต์แวร์ระบบหนึ่งจะพัฒนาโดยจัดกลุ่มการทำงานได้ทุกแบบ ถ้าจัดตามรูป 1a ก็เรียกได้ว่าซอฟต์แวร์นั้นเป็นระบบรวมศูนย์ การพัฒนาระบบจะใช้เครื่องมือในการพัฒนาบนเครื่องนี้เท่านั้น ในระบบเก่ามากหน่อยอาจจะใช้ภาษาโคบอลเขียน ระบบที่ใหม่ขึ้นมามักจะใช้ระบบฐานข้อมูลมาช่วยในการเก็บข้อมูล ส่วนที่ถือว่าเป็นของใหม่จริง ๆ ก็เป็นเรื่องที่เกี่ยวข้องกับการจัดการข้อมูลของระบบฐานข้อมูล เช่น การจัดการกับผู้ใช้หลาย ๆ คนที่ต้องการใช้ข้อมูลเดียวกัน การจัดการเรื่อง Transaction และความปลอดภัยของข้อมูล สิ่งต่าง ๆ เหล่านี้ไม่มีให้ในภาษาโคบอลหรือแม้แต่ภาษาที่เรียกว่าเป็นฐานข้อมูล เช่น dBASE บนเครื่องพีซี การจะเอาภาษาเหล่านี้มาเขียนเพื่อให้มีความสามารถตามที่กล่าวมาโปรเจ็กต์จะล้มเสียก่อน ดังนั้นวิธีการซื้อระบบฐานข้อมูลมาใช้เป็นส่วนประกอบในการพัฒนาระบบซอฟต์แวร์จึงเป็นเรื่องที่ต้องทำเลยทีเดียวถ้าอยากจะได้ความสามารถในการจัดการข้อมูลถึงขั้นนั้น

ส่วนที่ถือว่าเป็นของใหม่อีกอย่างหนึ่งคือ ระบบเครือข่าย ซึ่งเป็นตัวแปรสำคัญที่ทำให้เรามีอิสระในการจัดแบ่งงานมากขึ้น ทำให้เครื่องคอมพิวเตอร์ต่าง ๆ ทำงานร่วมกันได้ดีขึ้น เมื่อเครื่องพีซีที่ต่ออยู่กับระบบมีความสามารถมากขึ้นก็เอาไปช่วยกันทำงานได้ ทำให้ระบบไคล์เอ็นต์เซอร์ฟเวอร์นอกจากจะแบ่งตัวงานแล้วยังสามารถแบ่งว่าจะให้ใครทำได้อีกด้วย ใครถนัดอะไรก็ให้ทำงานนั้น ซึ่งเป็นที่มาของเรื่องไคล์เอ็นต์เซอร์ฟเวอร์นั่นเอง

รูปที่ 1d เป็นการแบ่งงานกันระหว่างเครื่องสองเครื่อง เครื่องหนึ่งเป็นเครื่องใหญ่ สามารถจัดการข้อมูลได้ดีก็จัดให้ทำหน้าที่บริการข้อมูล (Database Server) ไป อีกเครื่องหนึ่งเรียกว่าไคล์เอ็นต์ก็จะต้องทำงานส่วนที่เหลือ เช่น การติดต่อกับผู้ใช้ ทำการคำนวณ และออกรายงาน ถ้าถามว่าการที่มีระบบเครือข่ายเข้ามาเกี่ยวข้องมีผลต่อผู้พัฒนาระบบไหม? คำตอบก็คือ มีน้อยมาก เพราะมีการออกแบบให้มีการซ่อนการทำงานเครือข่ายไว้อย่างดี ถ้าเข้าใจการแบ่งงานที่กล่าวมาข้างต้นแล้ว การพัฒนาโปรแกรมบนระบบที่มีเครือข่ายอยู่ด้วยก็เป็นเรื่องที่ไม่ยากเกินไป

ปัจจุบันผู้ที่พัฒนาซอฟต์แวร์อยู่บนเครื่องพีซีก็มีโอกาสที่จะใช้ระบบฐาน-



รูปที่ 1 โมเดลการกระจายการทำงาน : ที่มา Gartner Group

ข้อมูลจริง ๆ ที่เหนือกว่า dBASE กันบ้างแล้ว เมื่อระบบ Novell LAN บนเครื่องพีซีเป็นที่นิยม และเครื่องพีซีมีความสามารถมากขึ้น ก็เริ่มมีการนำเอาฐานข้อมูลบนเครื่องมินิมาทำงานบนเครื่องพีซีที่รัน Novell NetWare และเมื่อไมโครซอฟต์ประกาศระบบปฏิบัติการวินโดวส์เอ็นที ซึ่งเป็นระบบปฏิบัติการบนเครื่องพีซีที่มีความสามารถไม่ด้อยไปกว่ายูนิกซ์ ก็มีฐานข้อมูลที่มีชื่อว่า SQL Server ซึ่งเป็นฐานข้อมูลที่มีความสามารถครบถ้วนและมีราคาน่าสนใจ

เหนือไคล์เอ็นต์เซอร์ฟเวอร์ยังมีไคล์เอ็นต์เซอร์ฟเวอร์

ระบบไคล์เอ็นต์เซอร์ฟเวอร์ที่ผมว่าไปข้างต้นนั้นนับว่าเป็นการจัดที่ดี เข้าใจง่าย มีการใช้งานมานานแล้ว แต่เมื่อไม่นานมานี้มีระบบชื่อไคล์เอ็นต์เซอร์ฟเวอร์เหมือนกัน แต่ดีกว่านั้นอีก อย่างว่าละครับ เหนือฟ้ายังมีฟ้าฉันใด เหนือไคล์เอ็นต์เซอร์ฟเวอร์ก็ยังมีไคล์เอ็นต์เซอร์ฟเวอร์ฉันนั้น

ไคล์เอ็นต์เซอร์ฟเวอร์แบบใหม่นี้เรียกได้ว่าเป็นฐานข้อมูลแบบพิเศษ ผู้ที่เริ่มนำมาใช้เป็นเจ้าแรกคือ Sybase แล้วตามมาด้วย SQL Server ของไมโครซอฟต์ ความสามารถพิเศษที่สำคัญที่สุดที่เพิ่มจากฐานข้อมูลแบบเก่าในรูปที่ 1d มีเพียงอย่างเดียวคือ การยอมให้ผู้ใช้ไป "เขียนโปรแกรม" เก็บไว้ในตัวฐานข้อมูลได้ กล่าวคือ ฐานข้อมูลแบบนี้จะไม่เพียงแต่รับรู้ภาษา SQL มาตรฐานเท่านั้น มันยังรับรู้ภาษาการเขียนโปรแกรมได้ด้วย เช่น มีคำสั่งในการแสดงผล คำสั่งในการวนรอบ คำสั่งในการตัดสินใจเลือก (IF, SelectCase) ท่านผู้อ่านอาจจะคิดว่าฐานข้อมูลตัวใหญ่กลับจะมาทำเลียนแบบโปรแกรม ดีเบสหรือยังไง เพราะว่าดีเบสก็เป็นฐานข้อมูลที่มีภาษาการเขียนโปรแกรมของตัวเองเหมือนกัน ประเด็นสำคัญอยู่ที่ว่า ในระบบไคล์เอ็นต์เซอร์ฟเวอร์แบบนี้จะมีเครือข่ายมาคั่นกลางการทำงานของแอพพลิเคชัน ดังในรูปที่ 1c ผลก็คือ เราจะมีความสามารถในการแบ่งงานระหว่างไคล์เอ็นต์และเซอร์ฟเวอร์ ถ้าเทียบกับระบบในรูปที่ 1d แล้ว ไคล์เอ็นต์เซอร์ฟเวอร์แบบใหม่จะแบ่งงานได้ละเอียดมากกว่า

บนฐานข้อมูลแบบนี้เราสามารถเขียนโมดูลขึ้นโดยใช้ภาษาของฐานข้อมูล เมื่อคอมพายล์เรียบร้อยจะเก็บอยู่ในตัวฐานข้อมูลเอง โมดูลเหล่านี้มีชื่อเรียกว่า Stored Procedure การมีโค้ดที่ประมวลผลการทำงานเก็บไว้ในฐานข้อมูลทำให้ไคล์เอ็นต์สามารถใช้โค้ดร่วมกันได้ และถ้าเราเข้าใจเรื่อง Object-Oriented เราก็จะสามารถสร้างฐานข้อมูลขององค์กรแบบ Object-Oriented ได้ ซึ่งจะช่วยผ่อนแรงผู้พัฒนาได้มากทีเดียว

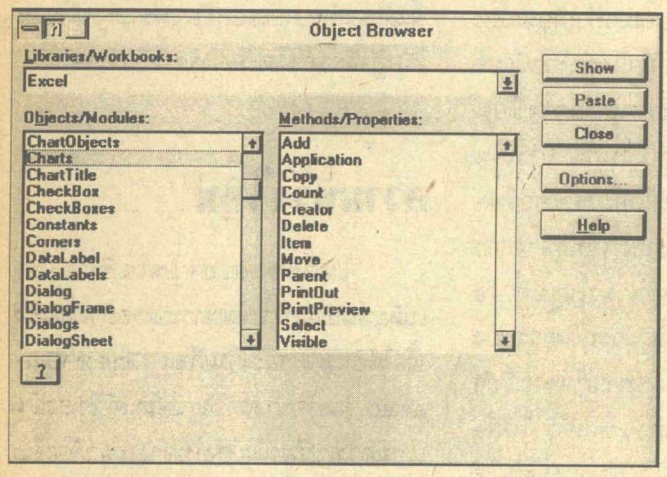
ในแง่ขององค์กรใหญ่ ๆ ที่กำลังปวดหัวกับการที่ระบบคอมพิวเตอร์ของฝ่ายต่าง ๆ มักจะใช้กฎเกณฑ์ในการทำงานไม่เหมือนกัน เช่น ฝ่ายขายของแต่ละเขตการขายอาจจะลดราคาของผลิตภัณฑ์โดยใช้หลักเกณฑ์ไม่เหมือนกัน ซึ่งถือว่าเป็นงานซ้ำซ้อนและมีผลเสียต่อองค์กร เราสามารถสร้างฐานข้อมูลผลิตภัณฑ์ที่รู้จัก "การคิดส่วนลด" อยู่ในตัวฐานข้อมูลเองตั้งเป็นฐานข้อมูลส่วนกลางให้ฝ่ายขายทั้งหมดมาขอใช้ ข้อมูลอื่นที่ไม่ใช่ของกลางก็ต่างคนต่างเก็บ การทำเช่นนี้เรียกว่าเป็นการสร้าง "กฎของธุรกิจ" (Bussiness Rules) ขึ้น แล้วฝังลงในฐานข้อมูลกลาง องค์กรไหนที่มีฐานข้อมูลแบบนี้แล้วละก็สบายเลยครับ เพราะทุกฝ่ายจะใช้กฎเดียวกันในเรื่องเดียวกันหมด

ความหวังใหม่

ในอนาคตเทคโนโลยีที่จะมาเปลี่ยนแปลงการพัฒนาระบบอย่างสิ้นเชิงนั้นได้ยินมาว่าจะเป็นเรื่อง Object-Oriented ผมว่าน่าจะจริง เพราะตัวของมันเองมีการเปลี่ยนแปลงไปจากแนวคิดเดิมมากคือ ส่วนที่เป็นของเก่าเอามาจัดใหม่มีประมาณแค่ 30-40 เปอร์เซ็นต์ของส่วนที่เป็นของใหม่ แต่ถ้าเป็นเรื่องอื่นที่กล่าวมาก่อนหน้าจะเป็นเรื่องเก่าเอามาจัดใหม่เสีย 60-70 เปอร์เซ็นต์ ตัวเลขข้างต้นเป็นตัวเลขคาดคะเนของผมเอง โดยเทียบจากสัดส่วนของเวลาที่ทำความเข้าใจว่าตรงไหนเป็นส่วนที่เอามาจัดใหม่ กับเวลาที่ใช้ในการทำความเข้าใจส่วนที่เป็นของใหม่หมด แต่สัดส่วนตรงนี้แหละที่ทำให้เกิดทั้งข้อดีและข้อเสียแก่ Object - Oriented

ข้อดีก็คือ เป็นความหวังใหม่ของชาวประชาที่จะมาปลดปล่อยให้เราหลุดพ้นจากความยากในการต่อจิ๊กซอ และเป็นวิธีการใหม่ที่ไม่เคยมีใครใช้มาก่อน

ข้อเสียก็คือ มันเข้าใจยาก อีก 60-70 เปอร์เซ็นต์ที่เหลือของ Object-Oriented เป็นเรื่องที่แตกต่างจากแนวความคิดในการเขียนโปรแกรมเดิมเป็นอย่างมาก การจะทำความเข้าใจส่วนนี้เทียบได้กับการพยายามฟังเพลงคลาสสิกให้ไพเราะฉันนั้น หลักฐานก็คือ หลังจากที่ใคร ๆ กล่าวถึงความสำเร็จในการนำไปใช้กันให้แซด ก็มีการวิเคราะห์ความล้มเหลวมาให้เห็นประปราย มีการ "ตั้งข้อสังเกต" เช่น ในเรื่องฐานข้อมูลก็มีผู้บอกว่าแอพพลิเคชันที่จำเป็นต้องใช้ฐานข้อมูล



รูปที่ 2
แสดงรายชื่อวัตถุบางส่วนที่มีอยู่ในโปรแกรมเอ็กเซล

แบบ Object-Oriented มีอยู่เพียงไม่กี่แบบ หรือมีการตัดสินใจไม่นำเอาโปรแกรมที่มีลักษณะเป็น Object-Oriented มาใช้ในองค์กรเนื่องจากทำให้ค่าใช้จ่ายในการเรียนรู้สูงเกินไป

ถึงตรงนี้ก็อย่าเพิ่งนึกว่าความหวังใหม่ของเราจะไปไม่ถึงดวงดาวนะครับ แนวโน้มที่จะเกิดขึ้นคือ ผู้ที่เข้าใจในเรื่อง Object-Oriented ก็จะอาศัยคุณสมบัติพิเศษของมันในการสร้าง "ส่วนประกอบ" ของระบบที่ใช้ได้ทั่วไปขึ้นมา ซึ่งการเขียนโปรแกรมของผู้พัฒนาระบบก็จะเป็นเหมือนการก่อสร้าง คือเป็นการนำเอาวัตถุมาต่อ ๆ กันแล้วจับให้ทำงานสัมพันธ์กันเท่านั้น

ดังเช่นที่เห็นได้ชัดคือ โปรแกรมวิชวลเบสิกของไมโครซอฟต์ ถ้าเราพัฒนาระบบโดยใช้โปรแกรมนี้ เราจะใช้ "ผลที่ได้" จากเรื่อง Object-Oriented แต่เราจะไม่ได้ใช้วิธีเชิงวัตถุในการพัฒนาระบบ ไมโครซอฟต์เองก็พยายามทำให้ทุก ๆ ผลิตภัณฑ์ของบริษัทเป็นวัตถุหมด เช่น โปรแกรมเอ็กเซลก็ประกอบด้วยวัตถุเอ็กเซล ซึ่งภายในวัตถุเอ็กเซลนี้ยังประกอบด้วยวัตถุเวอร์กชีตอีกชั้นหนึ่ง ในอนาคตระบบปฏิบัติการ Windows ของไมโครซอฟต์จะประกอบด้วยวัตถุทั้งหมด ดังนั้นในอนาคตเมื่อคุณติดตั้งซอฟต์แวร์ลงในเครื่องเสร็จ เครื่องของคุณก็จะประกอบไปด้วยวัตถุ วัตถุ และวัตถุ เต็มไปหมด ตอนนี้เริ่มฝึกหายใจเป็นวัตถุไว้บ้างก็ดีครับ

ส่งท้าย

สรุปแล้วผมมองว่าปัจจุบันเทคโนโลยีการพัฒนาซอฟต์แวร์ถึงแม้จะเปลี่ยนไปมาก แต่ยังเป็นเรื่องของการเอาของเก่ามาจัดใหม่ มีการแบ่งการทำงานชัดเจนขึ้น ซึ่งเป็นข้อดีที่ทำให้เราสามารถเลือกซื้อชิ้นส่วนซอฟต์แวร์มาประกอบเป็นระบบได้คล้ายกับการซื้อฮาร์ดแวร์ ผู้พัฒนาระบบก็ต้องทราบว่าระบบมีส่วนประกอบอะไรบ้าง แต่ละส่วนใช้ทำอะไร ส่วนไหนซื้อใช้ได้บ้าง ส่วนเรื่องของ Object-Oriented ในการพัฒนาซอฟต์แวร์เป็นสิ่งที่หลีกเลี่ยงไม่ได้ในอนาคตอย่างแน่นอน แต่จะมาในรูปใดก็ขึ้นกับว่าผู้พัฒนาทั่วไปเข้าใจ Object-Oriented มากน้อยแค่ไหน

เรียนรู้
CorelDRAW 4 อย่างรวดเร็ว
ด้วยการเรียนลัดด้วยภาพ

รหัสสั่งซื้อ 5120278
ราคา 280 บาท

คู่มือเรียนลัดด้วยภาพ
CorelDRAW 4
เรียนรู้ได้ด้วยตนเองจากภาพประกอบ
ด้วยคำอธิบายสั้น ๆ ทำให้เข้าใจส่วนต่าง ๆ
ของโปรแกรมและหน้าที่หลัก ๆ ของ
CorelDRAW ครบทุกส่วน

หาซื้อได้ตามร้านหนังสือชั้นนำทั่วประเทศ หรือที่ ซีเอ็ดบุ๊คเซ็นเตอร์ สาขาฟอร์จูนทาวน์ (ชั้นใต้ดิน) สาขาสีลมคอมเพล็กซ์(ชั้น 4) สาขาหลักสี่พลาซ่า(ชั้น 3) สาขาอิมพีเรียลเวิลด์สำโรง(ชั้นใต้ดิน) สาขาเซ็นทรัลซิตี้บางนา(ชั้น 3) สาขาเดอะมอลล์ บางกะปิ,บางแค(ชั้นG) สาขาฟิวเจอร์ปาร์ค บางแค(ชั้นใต้ดิน) และสามารถสั่งซื้อทางไปรษณีย์โดยตรงที่
บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)
800/43-45 ซ.ตระกูลสุข ถ.อโศกดินแดง เขตดินแดง กรุงเทพฯ 10400
โทร.248-6280-9 (อัตโนมัติ 20 สาย) แฟกซ์ 246-5999 โดยการส่งตั๋วแลกเงิน ธนาณัติหรือเช็คส่วนตัว สั่งจ่ายในนามบริษัทฯ กรณีเครดิตการ์ดโปรดระบุเงื่อนไขต่าง ๆตามการจ่ายเงินผ่านบัตร ค่าจัดส่ง 8 %พิเศษสำหรับสมาชิกวารสารของซีเอ็ด ลด 10% เพิ่มค่าลงทะเบียน 4 บาทค่าจัดส่งฟรีโดยระบุหมายเลขสมาชิก